02

Atelier 2

Atelier 2 : La qualité des données, un problème multiple

Atelier 2 : Normalisation

Gestion des variables catégorielles

Dans un jeu de données tabulaires, on retrouve fréquemment des données catégorielles. On distingue :

Les variables nominales (sans ordre) :

- **Type de véhicule** : Voiture, Camion, Moto, Vélo
- **Nationalité** : FR, EN, ES
- **Etat civil** : Célibataire, Marié, Divorcé, Veuf

Les variables ordinales (avec ordre) :

- **Niveau d'éducation** : secondaire, supérieur, doctorat
- **Niveau de risque** : Faible, Modéré, Elevé
- **Douleur ressentie** : Légère, Modérée, Sévère

One-hot Encoding

Nationalité
FR
FR
EN
ES
FR
EN

Nat-FR	Nat-EN	Nat-ES
1	0	0
1	0	0
0	1	0
0	0	1
1	0	0
0	1	0

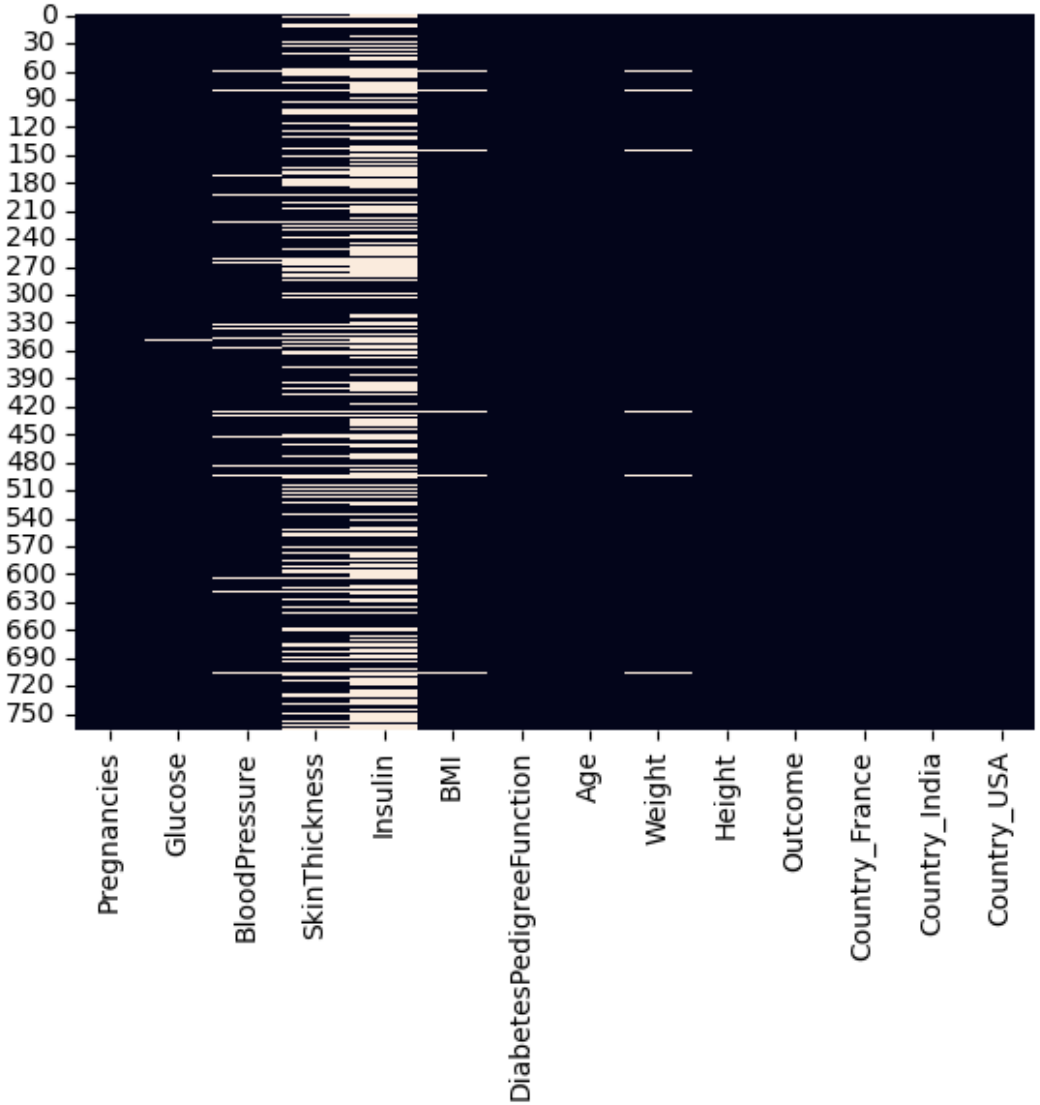
Label Encoding

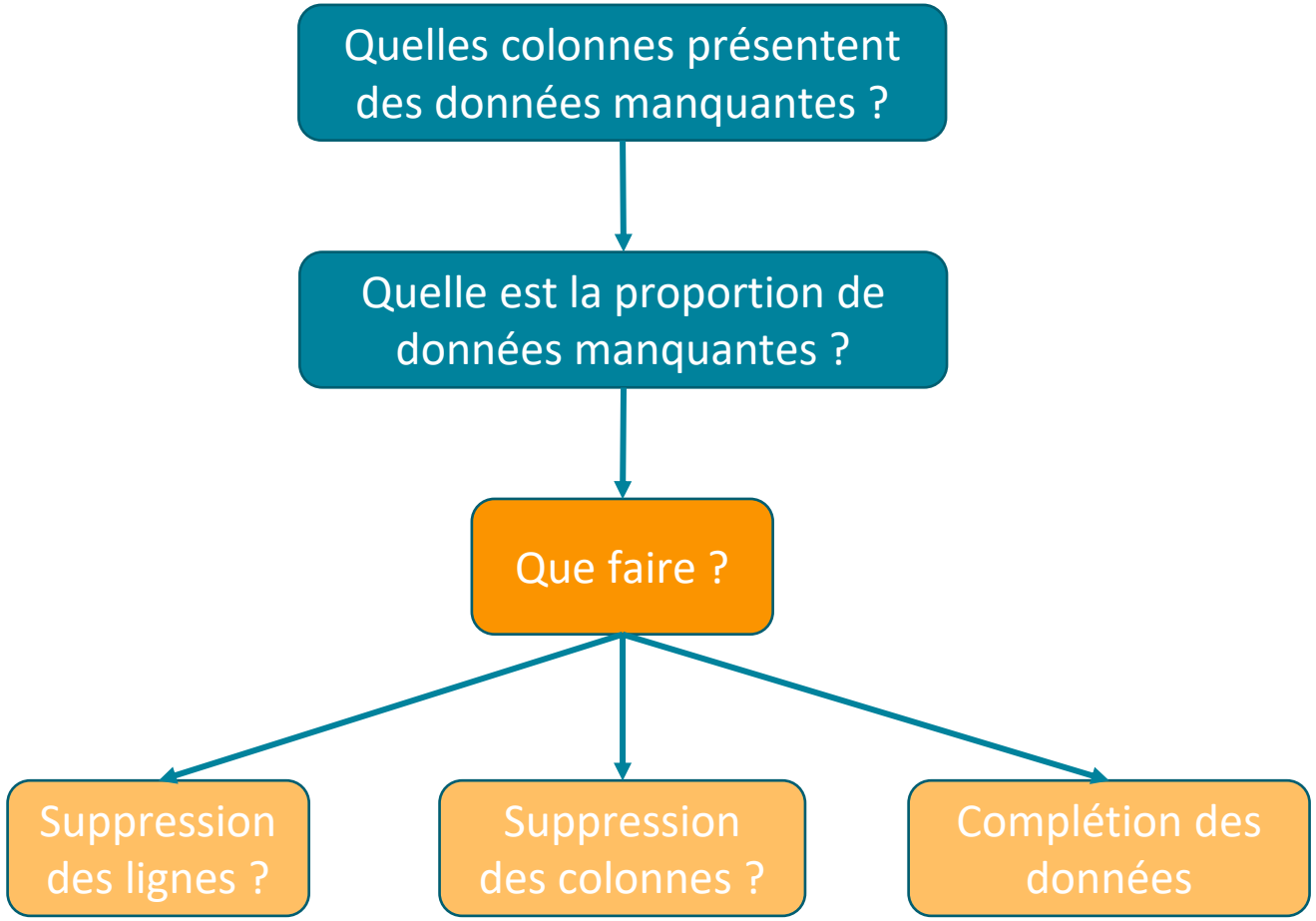
Douleur
Légère
Sévère
Modérée
Sévère
NoPain
Légère

Douleur
1
3
2
3
0
1

Atelier 2 : Gestion des données manquantes

Complétude / Données manquantes






```
seaborn.heatmap(df_encoded.isna(), cbar=False)
```

Atelier 2 : Gestion des données manquantes - Complétion

Complétude / Données manquantes

Remplacement par la moyenne

Feature B

10
X
25

X = (10 + 25) / 2

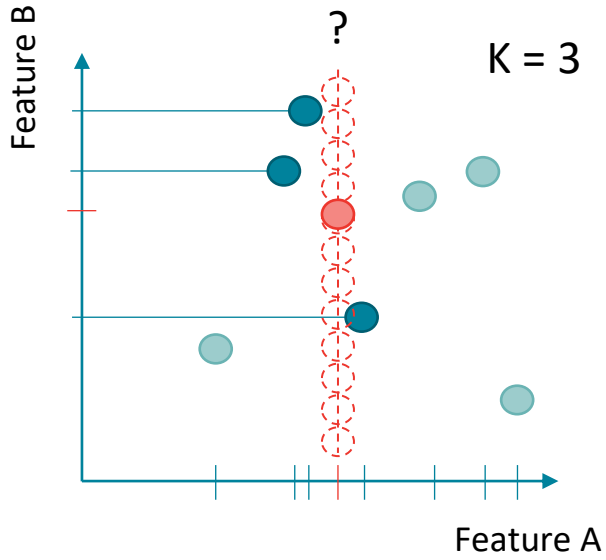
Interpolation linéaire (applicable sur des données ordonnées)

Feature A	Feature B
8	10
11	X
12	25

pente = (25 – 10) / (10 – 8)

X = 10 + pente * (11-8)

Complétion par plus proches voisins (K-NN)



Atelier 2 : Détection des outliers - Données aberrantes

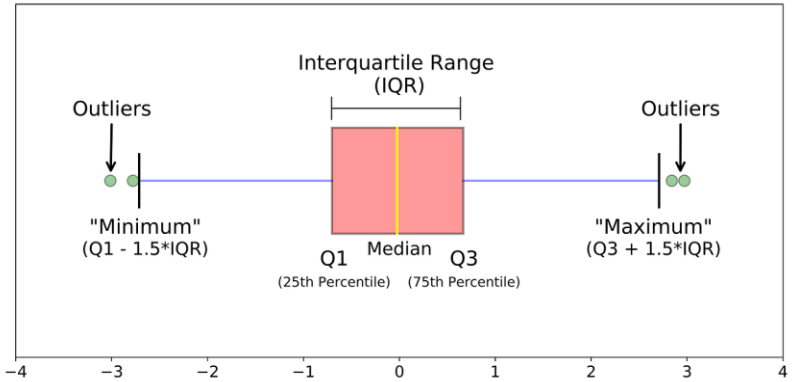
Gestion des outliers
Données aberrantes

Outlier (ou donnée aberrante) : Individu qui s'écarte significativement de la distribution initiale du jeu de données.

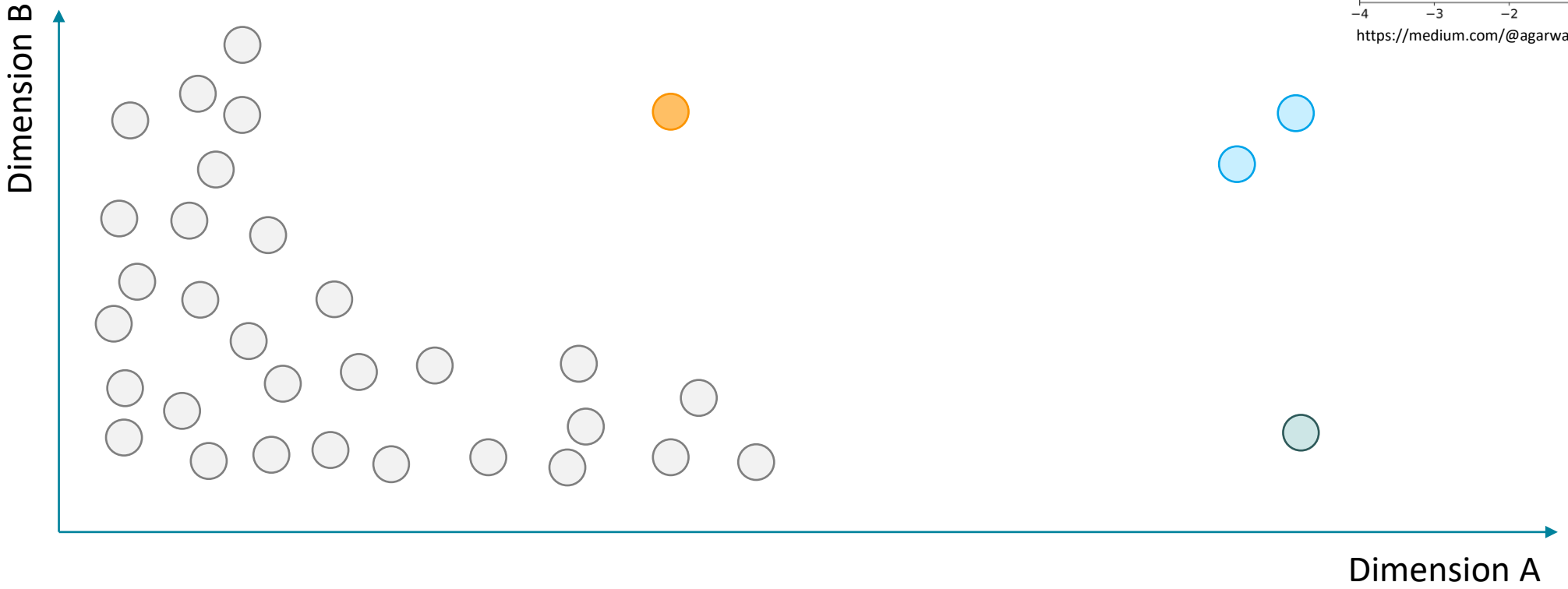
Certains algorithmes d'apprentissages sont très sensibles aux outliers.

On distingue :

- Les Outliers univariés, qui s'éloignent de la distribution sur une ou plusieurs dimensions de façon indépendante.
- Les Outliers multivariés, qui présentent une combinaison de features inhabituelle.



https://medium.com/@agarwal.vishal819/outlier-detection-with-boxplots-1b6757fafa21



Source : https://www.cuemath.com/data/outlier/

Atelier 2 : Équilibre de classes

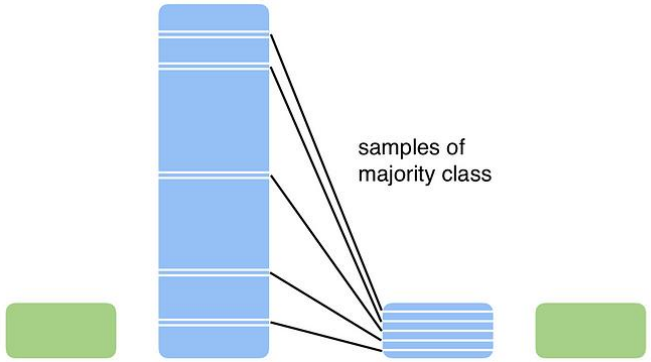
Équilibre des classes

La performance d'un modèle est guidée par la question suivante :
Le modèle a-t-il vu suffisamment d'exemples au cours de son entraînement ?

Hors, la majorité des jeux de données présentent un déséquilibre entre les différentes classes qu'il décrivent.

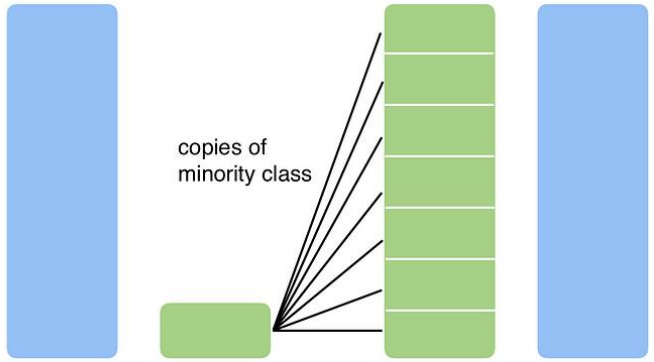
Un jeu de données idéal présente le même nombre d'individus pour chaque classe.

Undersampling



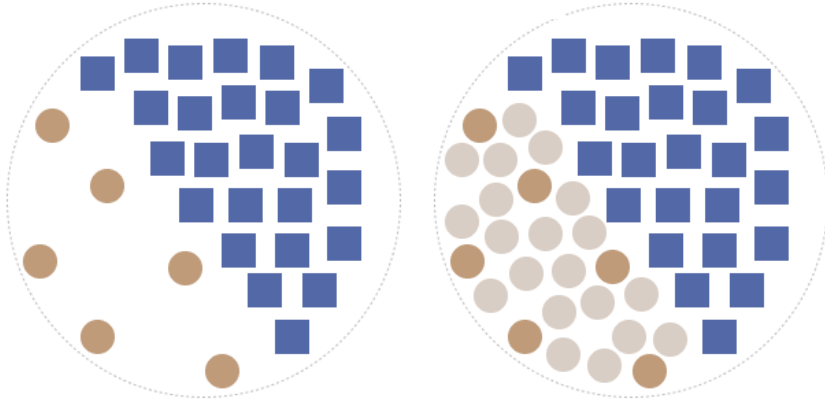
Jeu d'origine Jeu équilibré

Oversampling



Jeu d'origine Jeu équilibré

SMOTE



Jeu d'origine Jeu équilibré

Source : https://medium.com/strands-tech-corner/unbalanced-datasets-what-to-do-144e0552d9cd

https://dataknowsall.com/blog/imbalanced.html

Atelier 2 : Normalisation

Normalisation

Dans un jeu de données, il est possible d'avoir des features exprimées dans des échelles de données différentes

L'étape de normalisation permet de ramener les données sur une plage de valeurs comparable.
Cela permet de faciliter par la suite les calculs sur le jeu de données et de garantir une participation équitables des features à l'analyse et à l'apprentissage.

Pour normaliser les données on peut effectuer des transformations statistiques :

Normalisation Min-Max :

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Avec Python :

```
scaler = MinMaxScaler()
scaler.fit_transform(features)
```

Standardisation :

$$x' = \frac{x - \mu}{\sigma}$$

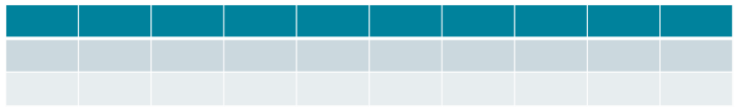
Avec Python :

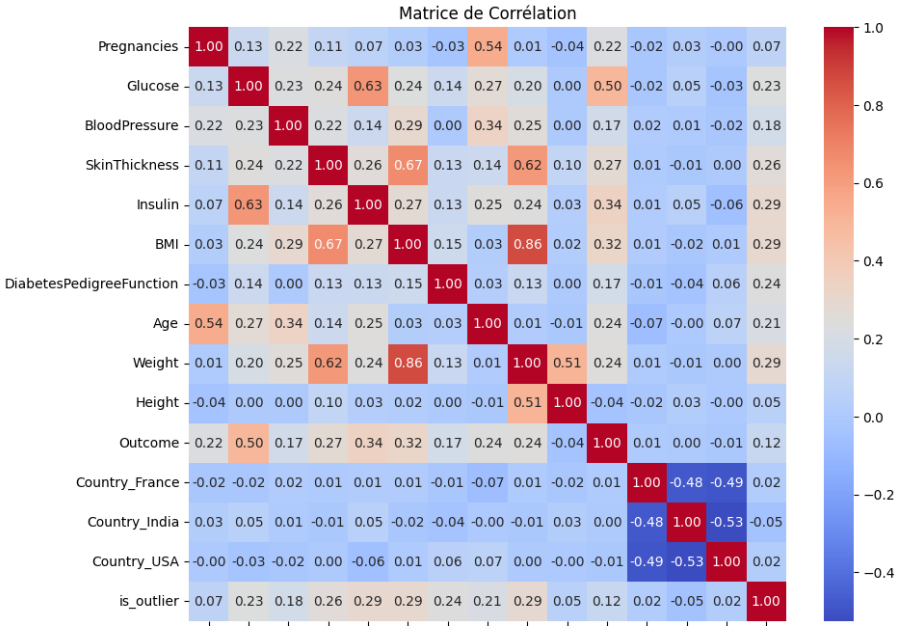
```
scaler = StandardScaler()
scaler.fit_transform(features)
```

Atelier 2 : Sélection des données

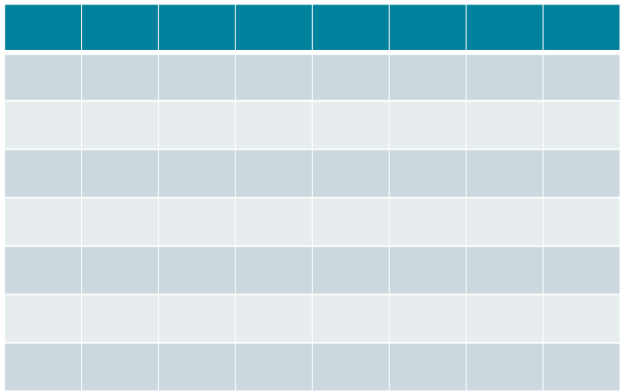
Pertinence et volume de données

N_individus << N_features



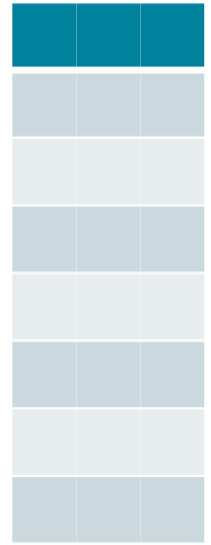


N_individus = N_features



Un outil pour la sélection de features : la matrice des corrélations

N_individus >> N_features



On cherche en général à avoir au moins 10 individus pour 1 feature.